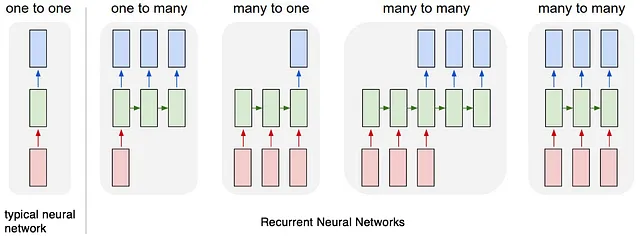
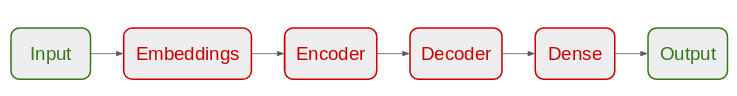
İNGİLİZCE -ALMANCA ÇEVİRİ UYGULAMASI:

Uygulama için gerekli modeli oluştururken LSTM modelini kullandım. LSTM, RNN modellerinden bir tanesi. RNN daha sınırlı bir hafızaya sahipken, LSTM daha uzun süreli hafızaya sahip. Ayrıca LSTM bilginin hatırlanacağına veya unutulacağına da süreç içerisinde karar veriyor.



Machine translation many-to-many process kullanmayı gerektiriyor. Aynı zamanda model katmaları aşağıdaki mimariye sahip olmalı.



Bu mimaride input’lar tokenization ile sayısal değerlere çevrilmiş text’ler. Embedding katmanı input’lardaki kelimelerin yerlerini de göz önünde bulundurarak yeni vektörler oluşturur. Encoder ingilizce girdileri kullanarak bir context yaratır. Decoder bu context’i kullnarak gelen girdileri decode eder. Dense katmanı almanca kelimelere uygun indexed text vektörünü yaratır.

Encoder ve decoder katmanlarının karar verebilmeleri için sadece o timestepte gelen kelimeyi değil öncekileri de göz önünde bulundurması gerekir bunun için Bu katmanlarda LSTM kullandım.

Sadece önceki kelimelere değil, daha sonra gelecek kelimelere de göre karar veren bir yapının daha doğru sonuçlar vereceğini düşündüm. İki katmanın Bidirectional olduğu ikinci bir model daha oluşturdum. Adam optimizer’ı kıllandığım bu modellerden ikincisi biraz daha iyi bir sonuç verdi.

model1:

loss: 0.2293 - accuracy: 0.9178 - val\_loss: 2.3676 - val\_accuracy: 0.7438

model2:

loss: 0.1319 - accuracy: 0.9232 - val\_loss: 2.7542 - val\_accuracy: 0.7379

Daha sonra bu iki modeli tekrar SGD optimizer’ı ile çalıştırdım. Adam’a göre aldığım sonuçlar daha kötüydü. Bu sonucun local min nedeniyle olabileceğini düşünüyorum.

model3:

loss: 2.3362 - accuracy: 0.6544 - val\_loss: 2.3269 - val\_accuracy: 0.6543

Model4:

loss: 2.3013 - accuracy: 0.6544 - val\_loss: 2.3027 - val\_accuracy: 0.6543